

Cercador d'imatge optimitzat amb OpenMP

Persistència de dades

Quan carreguem un conjunt d'imatges, aquestes van a parar en el mateix directori on està l'executable. Allà mateix es creen dues carpetes, 'hist/' i 'img/'. Les imatges aniran a 'img/' amb el seu nom acordat (img_XXXXXX.jpg) i els histogrames corresponents a cadascuna de les imatges aniran a 'hist/' amb el seu nom acordat també (hist_XXXXXX.xml).

Quan s'executa el programa el primer que fa es comprovar si existeix la carpeta 'hist/', en cas afirmatiu carrega tots els histogrames de disc a memòria RAM.

Carregar imatges

Per tal de carregar un conjunt d'imatges, seguim el següent algoritme:

- Copiem la imatge (ús de la funció 'sendfile') al directori /img amb el seu nom corresponent.
- Obrim la imatge, creem el seu histograma, i el guardem a /hist.

Execució del programa

El programa és ben senzill, hi ha una comanda per carregar un conjunt d'imatges en la barra del menú, i una vegada la base de dades té carregada un conjunt d'imatges podem buscar una imatge, i un cop escollida una imatge es realitza la cerca.

La sortida del programa és graella de 9 imatges, ordenades per semblança a la imatge d'entrada. La més semblant és la de la cantonada superior esquerra, i a partir d'allà va d'esquerra a dreta i d'adalt abaix.

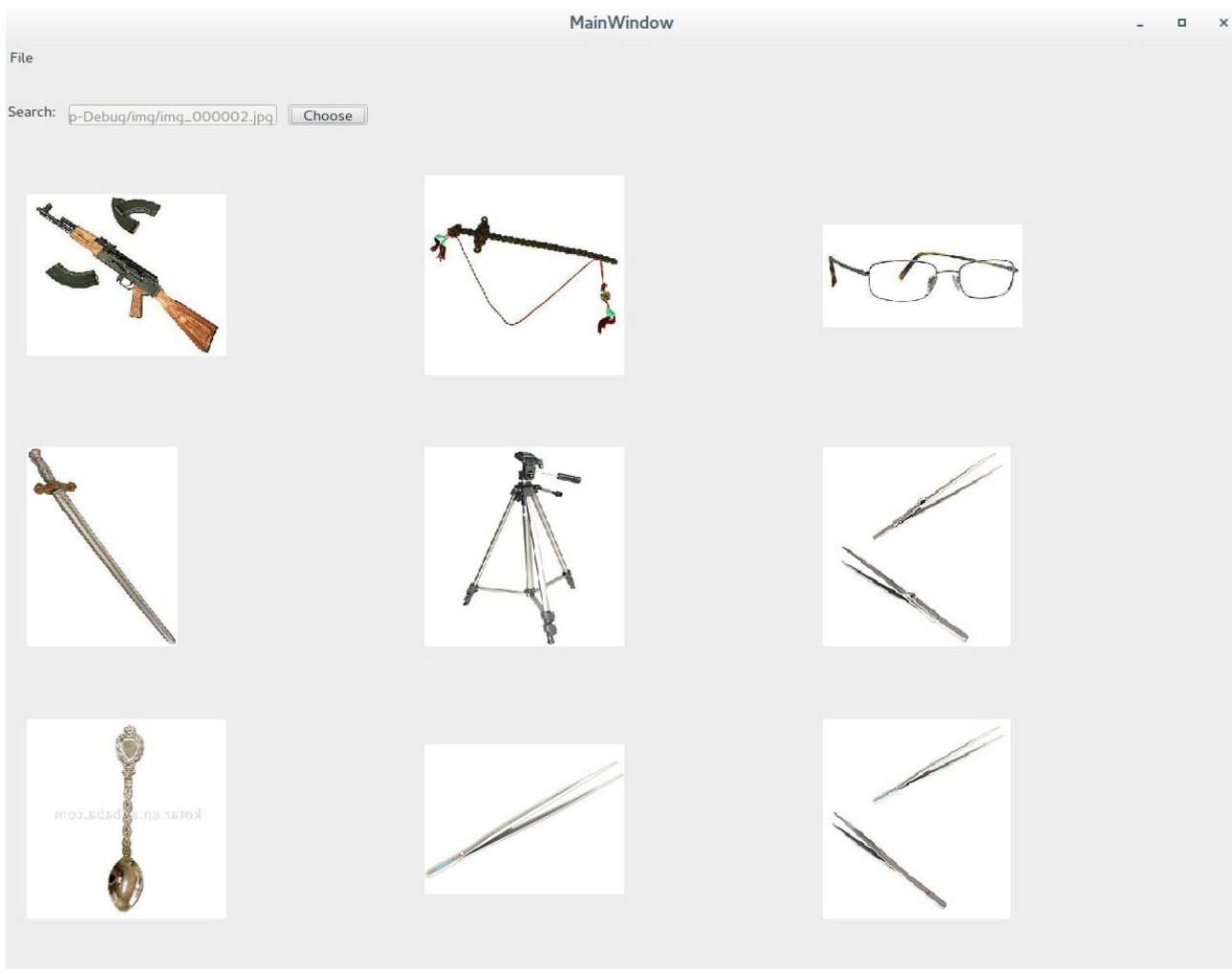


Figura 1 – Resultat d'execució del search.

Per escollir un conjunt d'imatges, s'ha d'escollir un directori i no un fitxer com recomanava a l'enunciat.

OpenMP

Primer vam fer el programa per a que s'executés en seqüencial, tenint en compte que el següent pas seria paral·lelitzar el codi. Realment el més ràpid va ser paral·lelitzar el codi. Les zones que hem paral·lelitzat han sigut les següents:

- Carregar els histogrames de disc a memòria RAM.
- Carregar un conjunt d'imatges a la base de dades.
- Fer la cerca d'una imatge.

Més baix, en les proves d'execució expliquem les conclusions a les que hem arribat, i amb quina sentència d'OpenMP hem resolt la concurrència i perquè.

Proves d'execució

Les proves les hem fet amb un conjunt de 15000 i 30000 imatges. Per cada conjunt d'imatges, hem testejat (tot en segons):

- loadHist: temps que tarda en carregar els histogrames guardats en disc a memòria RAM.
- insertarImatges: temps que tarda en realitzar l'algoritme de la secció '**Carregar imatges**'.
- search: temps que tarda en fer una cerca.

Aquests han sigut els resultats:

Seqüencial:

	loadHist	insertImages	search
15.000	1.0253	63.7537	0.0571829
30.000	1.99816	158.484	0.0575576

En la part de concurrència hem provat d'implementar l'algoritme de 'insertarImages' de dues formes, un amb « #pragma omp parallel for » i l'altre amb tasques.

Les tasques que hem fet són dues:

- Copiar les imatges a la carpeta '/img'.
- Crear i guardar els histogrames a '/hist'.

Concurrent amb for:

	loadHist	insertImages	search
15000	0.802993	15.6934	0.0161003
30000	1.82574	31.6024	0.0259355

Concurrent amb tasques:

	insertImages
15000	15.5083
30000	29.9232

La pràctica l'hem entregat amb tasques, tot i que en temps d'execució no es veuen grans canvis. Hem cregut que és millor posar tasques, ja que les tasques ha fer són pesades i d'aquesta manera distribuïm millor les tasques entre els fils. Si ho tinguessim amb « #pragma omp parallel for », la càrrega seria la mateixa per a tots els fils, i potser al final de l'execució ens podem trobar fils que hagin terminat i altres encara estiguessin fent feina.

Problemes

En general no hem tingut molts problemes amb aquesta pràctica, tot i que recalcarem els tres problemes principals que hem tingut.

1. Per començar vam estar més temps integrant opencv i openmp amb qt, que no pas programant la pràctica en si.
2. Un cop vam finalitzar la pràctica en seqüencial i ens funcionava perfectament en els ordinadors de l'aula IA, vam veure que en el nostre portàtil la funció de comparar histograma, no retornava el mateix que els ordinadors de la universitat.
3. Quan visualitzavem les imatges de la cerca, com les llegiem amb openCV, però les mostravem amb qt, teniem el problema que es mostravem les imatges amb els canals invertits. Resulta que openCV treballa amb BGR i qt amb RGB, per tal de solucionar-ho vam fer la crida a la funció `cvtColor`.